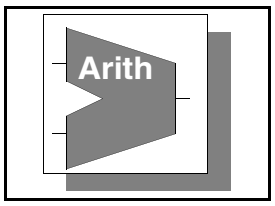
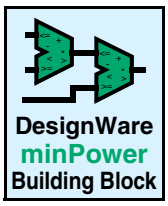




DW_div_pipe

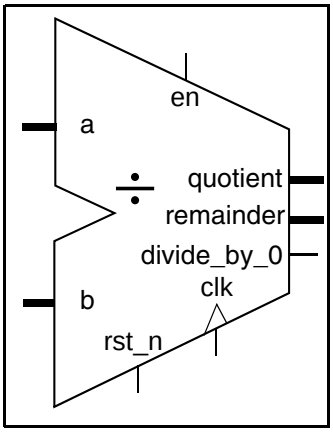
Stallable Pipelined Divider

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

Revision History

- Re-uses the DW_div component and adds pipeline structures
- Parameterized word length
- Parameterized unsigned and signed data operation
- Parameterized number of pipeline stages
- Parameterized stall mode (stallable or non-stallable)
- Parameterized reset mode (no reset, asynchronous or synchronous reset)
- Automatic pipeline retiming
- Provides minPower benefits (see Table 1-3 on page 3)



Description

DW_div_pipe is a universal stallable pipelined divider with optional low-power benefits. It contains the DW_div component to perform the division, and additional pipelining register logic and timing.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
clk	1 bit	Input	Input clock
rst_n	1 bit	Input	Reset, active-low (not used if parameter *rst_mode* = 0)
en	1 bit	Input	Register enable, active high (used only if parameter *stall_mode* = 1) ■ 0: Stall ■ 1: Enable register
a	*a_width* bits	Input	Dividend
b	*b_width* bits	Input	Divisor
quotient	*a_width* bits	Output	Quotient `a` / `b`
remainder	*b_width* bits	Output	Remainder
divide_by_0	1 bit	Output	Indicates if `b` equals zero

Table 1-2 Parameter Description

Parameter	Values	Description
a_width	≥ 2 Default: None	Word length of `a`
b_width	≥ 2 Default: None	Word length of `b`
tc_mode	0 or 1 Default: 0	Two's complement control ■ 0: Unsigned ■ 1: Signed
rem_mode	0 or 1 Default: 1	Remainder output control ■ 0: Modulus ■ 1: Remainder
num_stages	≥ 2 Default: 2	Number of pipeline stages
stall_mode	0 or 1 Default: 1	Stall mode ■ 0: Non-stallable ■ 1: Stallable
rst_mode	0 to 2 Default: 1	Reset mode ■ 0: No reset ■ 1: Asynchronous reset ■ 2: Synchronous reset
op_iso_mode	0 to 4 Default: 0	Operand isolation mode (controls datapath gating for minPower flow) Allows you to set the style of minPower datapath gating for this module ■ 0: Use the DW_lp_op_iso_mode[a] synthesis variable ■ 1: 'none' ■ 2: 'and' ■ 3: 'or' ■ 4: Preferred gating style: 'and' For details about enabling minPower datapath gating for this component, see "Enabling minPower" on page 7.

a. The DW_lp_op_iso_mode synthesis variable is available only in Design Compiler.
DW_lp_op_iso_mode sets a global style of datapath gating. To use the global style, set *op_iso_mode* to '0', Note that If the *op_iso_mode* parameter is set to '0' and DW_lp_op_iso_mode is either not set or set to 0', then no datapath gating is inserted for this component.

Table 1-3 Synthesis Implementations

Implementation Name	Implementation	License Feature Required
str	Pipelined str synthesis model	DesignWare
lpwr[a]	Low Power synthesis model	■ DesignWare (P-2019.03 and later) ■ DesignWare-LP (before P-2019.03)

a. Requires that you enable minPower; for details, see "Enabling minPower" on page 7.
When minPower is enabled, the lpwr implementation is always chosen during synthesis.

Table 1-4 Simulation Models

Model	Function
DW02.DW_DIV_PIPE_CFG_SIM	Design unit name for VHDL simulation
dw/dw02/src/DW_div_pipe_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_div_pipe.v	Verilog simulation model source code

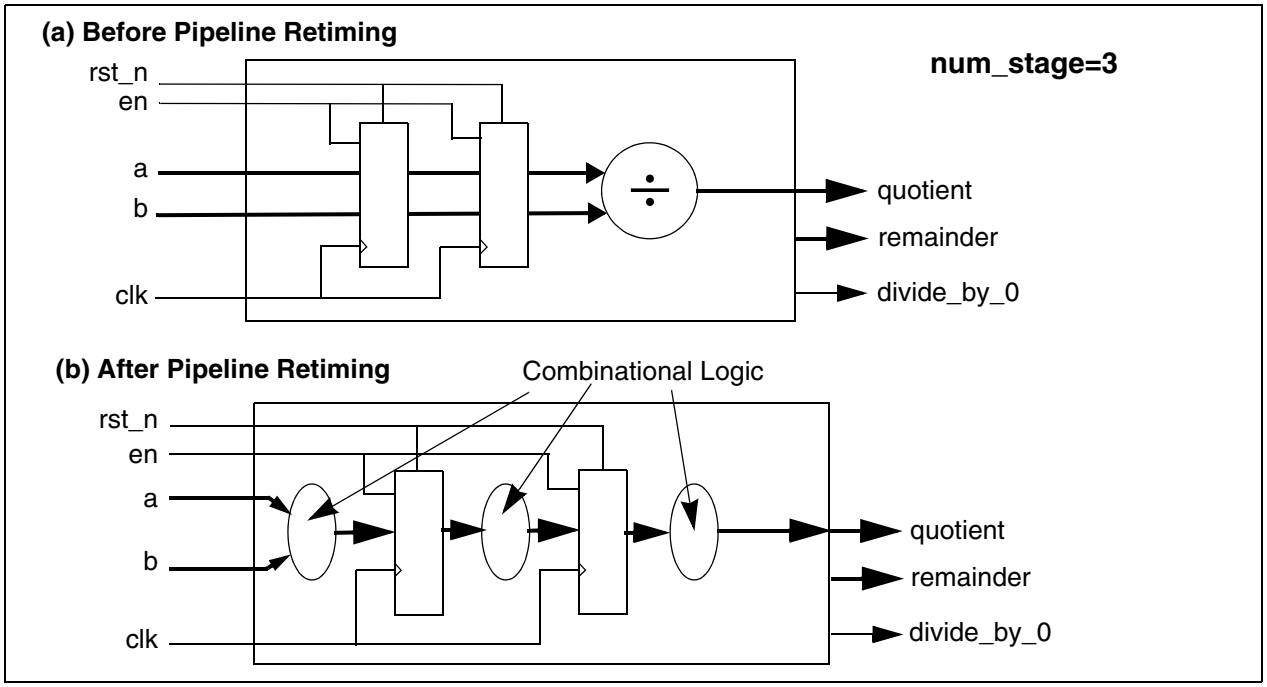
DW_div_pipe divides the operands `a` by `b` to produce a `quotient` and a `remainder`, if selected through a parameter, with a latency of *num_stages*–1 clock (`clk`) cycles. The parameter *tc_mode* determines whether the input and output data is interpreted as unsigned (*tc_mode* = 0) or signed (*tc_mode* = 1) numbers.

Attention A divide by zero warning message is generated each time the `b` input changes to the value zero. This warning can be disabled for Verilog simulations by defining the Verilog macro, DW_SUPPRESS_WARN, either in the test bench or on the simulator command line (such as, `+define+DW_SUPPRESS_WARN+`).

The DW_div_pipe incorporates the DW_div component to perform the division. For more information on the DW_div component, see *DW_div Combinational Divider*.

Automatic pipeline retiming ensures optimal placement of pipeline registers within the divider to achieve maximum throughput. The pipeline can be stalled by setting the load enable signal `en` low(*stall_mode* = 1). The pipeline registers can either have no reset (*rst_mode* = 0) or an asychronous (*rst_mode* = 1) or synchronous reset (*rst_mode* = 2) connected to the reset signal `rst_n`.

Figure 1-1 Pipeline Retiming



Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If values other than 1 or 0 are present on a clock port, the following message is displayed:

```
WARNING: <instance_path>.<clock_name>_monitor:
    at time = <timestamp>, Detected unknown value, x, on <clock_name> input.
```

To suppress only this warning message for all DWBB components, use the following macro:

- Define the DW_DISABLE_CLK_MONITOR macro. You can define this macro in the following ways:
  - Specify the Verilog preprocessing macro in Verilog code:

```
`define DW_DISABLE_CLK_MONITOR
```

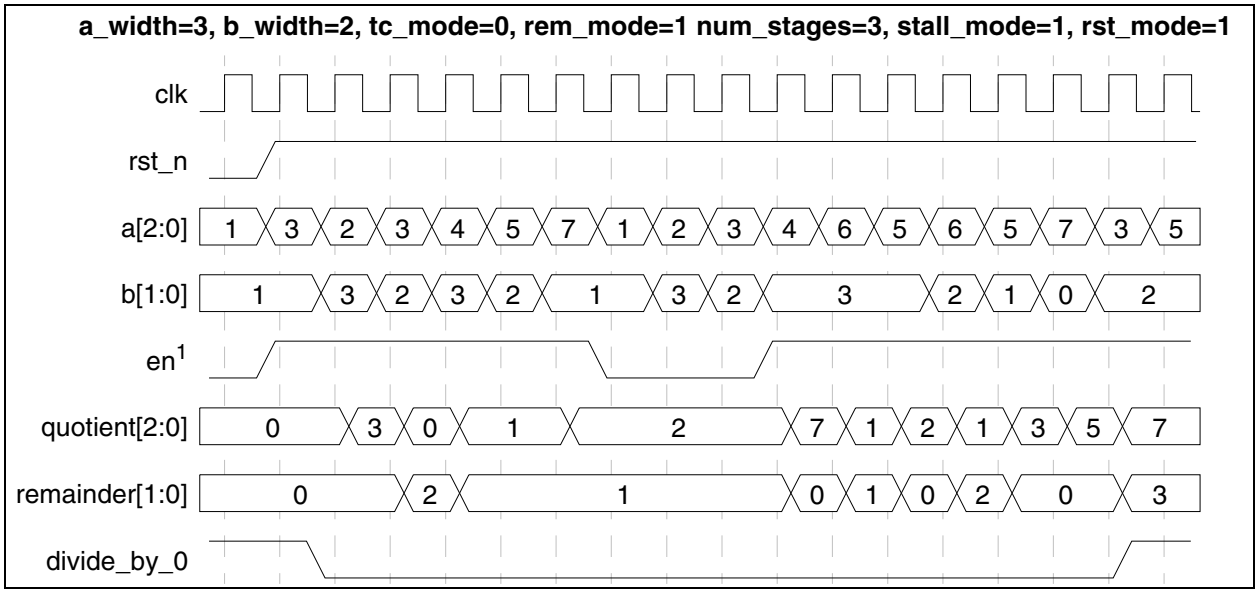
  - Or, include a command line option to the simulator, such as:

`+define+DW_DISABLE_CLK_MONITOR` (which is used for the Synopsys VCS simulator)

This message is also suppressed using the DW_SUPPRESS_WARN macro explained earlier.

Timing Waveforms

Figure 1-2 Waveform 1



[1]If parameter stall_mode=0, then pin en has no effect.

Figure 1-3 Waveform 2

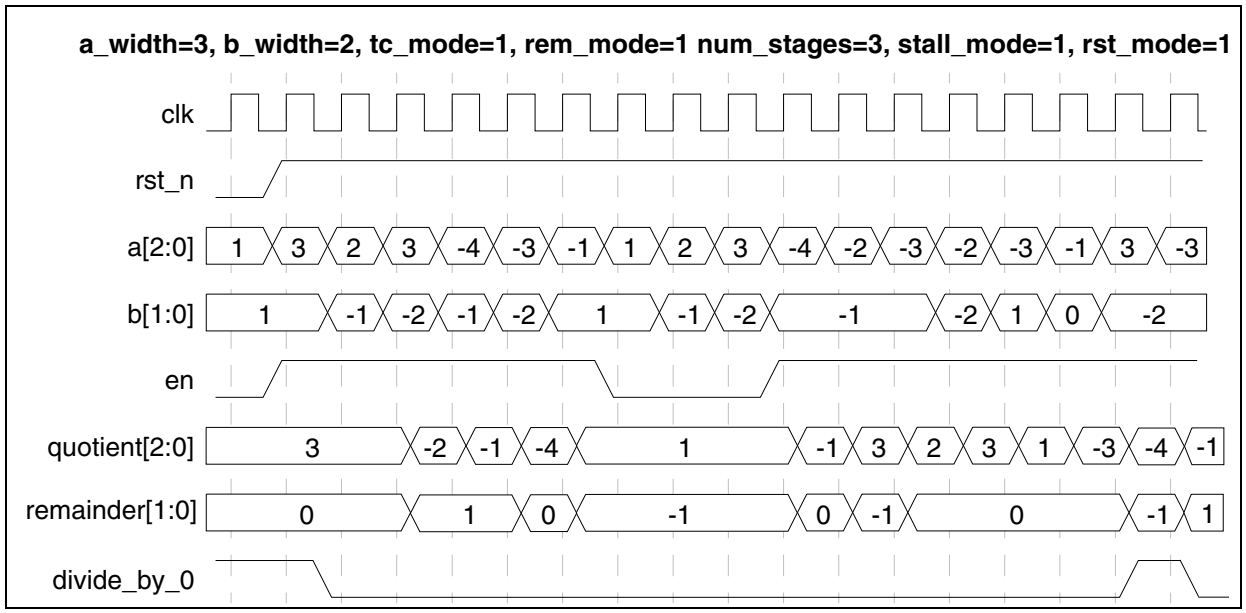


Figure 1-4 Waveform 3

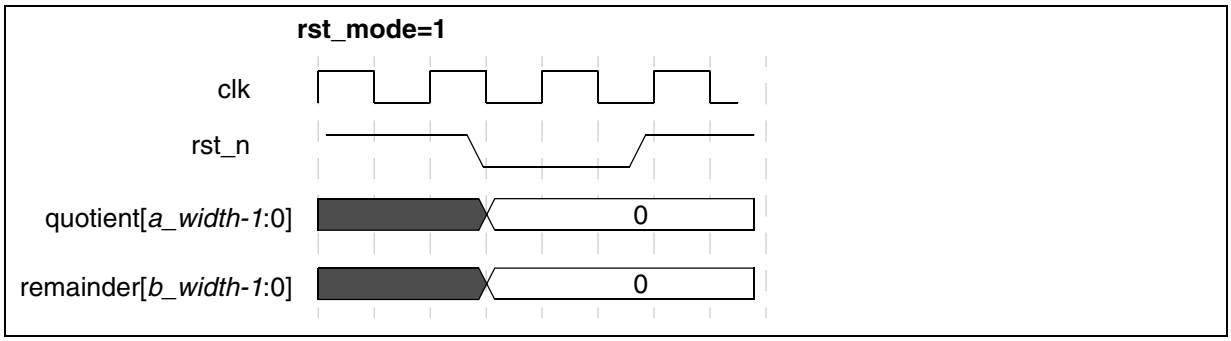
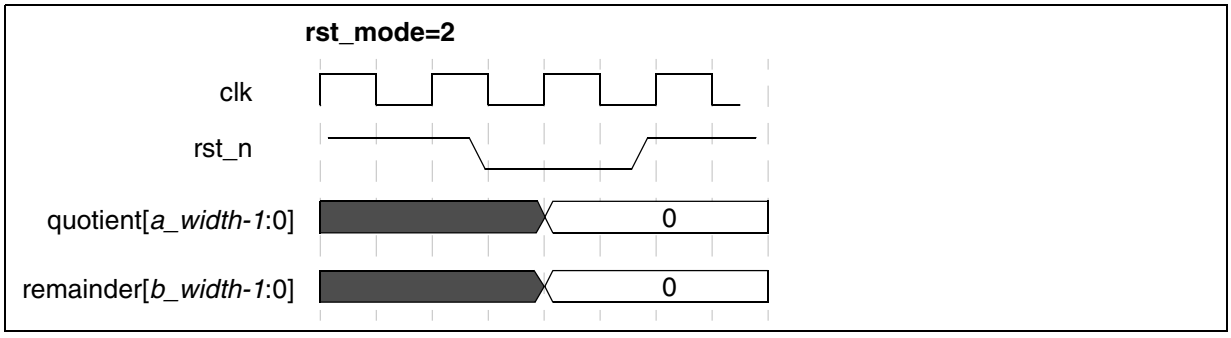


Figure 1-5 Waveform 4



Enabling minPower

You can instantiate this component without enabling minPower, but to achieve datapath gating power savings, you must enable minPower optimization, as follows:

- Design Compiler
  - Version P-2019.03 and later:

    ```
    set power_enable_minpower true
    ```

  - Before version P-2019.03 (requires the DesignWare-LP license feature):

    ```
    set synthetic_library {dw_foundation.sldb dw_minpower.sldb}
    set link_library {* $target_library $synthetic_library}
    ```

- Fusion Compiler

  Optimization for minPower is enabled as part of the total_power metric setting. To enable the total_power metric, use the following:

  ```
  set_qor_strategy -stage synthesis -metric total_power
  ```

Related Topics

- Math – Arithmetic Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE, DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DW_Foundation_comp_arith.all;

entity DW_div_pipe_inst is
  generic (inst_a_width    : POSITIVE := 8;  inst_b_width     : POSITIVE := 8;
           inst_tc_mode    : NATURAL := 0;   inst_rem_mode    : NATURAL := 1;
           inst_num_stages : POSITIVE := 2;  inst_stall_mode  : NATURAL := 1;
           inst_rst_mode   : NATURAL := 1;   inst_op_iso_mode : NATURAL := 0 );
  port (inst_clk         : in std_logic;
        inst_rst_n       : in std_logic;
        inst_en          : in std_logic;
        inst_a           : in std_logic_vector(inst_a_width-1 downto 0);
        inst_b           : in std_logic_vector(inst_b_width-1 downto 0);
        quotient_inst    : out std_logic_vector(inst_a_width-1 downto 0);
        remainder_inst   : out std_logic_vector(inst_b_width-1 downto 0);
        divide_by_0_inst : out std_logic );
end DW_div_pipe_inst;

architecture inst of DW_div_pipe_inst is
begin
  -- Instance of DW_div_pipe
  U1 : DW_div_pipe
    generic map (a_width => inst_a_width,   b_width => inst_b_width,
                 tc_mode => inst_tc_mode,   rem_mode => inst_rem_mode,
                 num_stages => inst_num_stages,  stall_mode => inst_stall_mode,
                 rst_mode => inst_rst_mode,   op_iso_mode => inst_op_iso_mode )
    port map (clk => inst_clk,   rst_n => inst_rst_n,   en => inst_en,
              a => inst_a,   b => inst_b,
              quotient => quotient_inst,   remainder => remainder_inst,
              divide_by_0 => divide_by_0_inst );
end inst;

-- Configuration for use with VSS simulator
-- pragma translate_off
configuration DW_div_pipe_inst_cfg_inst of DW_div_pipe_inst is
  for inst
  end for; -- inst
end DW_div_pipe_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_div_pipe_inst(inst_clk, inst_rst_n, inst_en, inst_a, inst_b,
                        quotient_inst, remainder_inst, divide_by_0_inst );

  parameter inst_a_width = 8;
  parameter inst_b_width = 8;
  parameter inst_tc_mode = 0;
  parameter inst_rem_mode = 1;
  parameter inst_num_stages = 2;
  parameter inst_stall_mode = 1;
  parameter inst_rst_mode = 1;
  parameter inst_op_iso_mode = 0;

  input inst_clk;
  input inst_rst_n;
  input inst_en;
  input [inst_a_width-1 : 0] inst_a;
  input [inst_b_width-1 : 0] inst_b;
  output [inst_a_width-1 : 0] quotient_inst;
  output [inst_b_width-1 : 0] remainder_inst;
  output divide_by_0_inst;

  // Instance of DW_div_pipe
  DW_div_pipe #(inst_a_width,   inst_b_width,   inst_tc_mode,  inst_rem_mode,
                inst_num_stages,   inst_stall_mode,   inst_rst_mode,
  inst_op_iso_mode)
    U1 (.clk(inst_clk),   .rst_n(inst_rst_n),   .en(inst_en),
        .a(inst_a),   .b(inst_b),   .quotient(quotient_inst),
        .remainder(remainder_inst),   .divide_by_0(divide_by_0_inst) );
endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Adjusted content and title of "Suppressing Warning Messages During Verilog Simulation" on page 4 and added the DW_SUPPRESS_WARN macro
October 2019	DWBB_201903.5	■ Added the "Disabling Clock Monitor Messages" section
March 2019	DWBB_201903.0	■ Clarified the op_iso_mode parameter in Table 1-2 on page 2 ■ Clarified license requirements in Table 1-3 on page 3 ■ Added "Enabling minPower" on page 7 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com